Task Priority-based Inverse Kinematics

Notes by: Jun Young Kim

Basic definitions

Given tasks are defined as follows:

$$(\boldsymbol{x}_1, \boldsymbol{J}_1),\ (\boldsymbol{x}_2, \boldsymbol{J}_2),\ \cdots,\ (\boldsymbol{x}_k, \boldsymbol{J}_k) \tag{1}$$

tasks could represent the end-effector position and/or orientation.

All n tasks can be done simultaneously by utilizing the pseudo-inverse of the augmented Jacobian matrix by stacking the n tasks together as follows:

$$\dot{\boldsymbol{q}} = \underbrace{\begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \\ \vdots \\ \boldsymbol{J}_n \end{bmatrix}^{\#}}_{\hat{\boldsymbol{J}}_n^{\#}} \underbrace{\begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \vdots \\ \dot{\boldsymbol{x}}_n \end{bmatrix}}_{\hat{\boldsymbol{x}}_n} \tag{2}$$

where $\{\hat{\cdot}\}$ means stacked augmented Jacobian and stacked tasks, $\{\cdot\}^{\#}$ denotes pseudo-inverse. However, it is hard to use directly because two or more tasks might conflict with each other. In this case, some issues might arise, such as 1) augmented Jacobian (2) will drop rank and become singular, 2) pseudo-inverse of (2) can create unreasonably high joint velocities. We could solve this issue in two approaches, first we potentially deal with singularities by using a damped pseudo-inverse as follows:

$$\boldsymbol{J}^{\#\lambda} = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T + \lambda^2\boldsymbol{I})^{-1} \tag{3}$$

at the cost of inaccurate control of tasks. λ is a positive scalar called the damping factor. Another way is to utilize the task-priority framework.

Several Approaches

Y Nakamura et. al [1]

Following the [2], task Priority control of redundant manipulators has been studied in several papers; in one of the originals [1], task-priority based control of two tasks is defined as:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\#}\dot{\boldsymbol{x}}_1 + (\boldsymbol{J}_2\boldsymbol{N}_1)^{\#}(\dot{\boldsymbol{x}}_2 - \boldsymbol{J}_2\boldsymbol{J}_1^{\#}\dot{\boldsymbol{x}}_1) \tag{4}$$

where $\boldsymbol{N}_i = \boldsymbol{I} - \boldsymbol{J}_i^{\#}\boldsymbol{J}_i$ is the null-space projection matrix of task Jacobian. A lower index number means higher priority, and the framework is generalized, recursively in n tasks by [3].

Siciliano and Slotine [3]

Siciliano proposed a general framework for managing multiple tasks in highly redundant robotics systems, which will guarantee the execution of higher priority. First, consider a generic i-th task is defined by:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i\dot{\boldsymbol{q}} \tag{5}$$

where $\boldsymbol{q}$ is the joint configuration vector.

$$\begin{aligned} \dot{\boldsymbol{q}}_i &= \dot{\boldsymbol{q}}_{i-1} + (\boldsymbol{J}_i\hat{\boldsymbol{N}}_{i-1})^{\#\lambda}(\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i\dot{\boldsymbol{q}}_{i-1}) & (6)\\ \dot{\boldsymbol{q}} &= \dot{\boldsymbol{q}}_n & (7)\\ \text{where}\quad \hat{\boldsymbol{N}}_i &= \boldsymbol{I} - \hat{\boldsymbol{J}}_i^{\#}\hat{\boldsymbol{J}}_i, \quad \hat{\boldsymbol{N}}_0 = \boldsymbol{I} & (8)\\ \dot{\boldsymbol{q}}_1 &= \boldsymbol{J}_1^{\#}\dot{\boldsymbol{x}}_1, \qquad \dot{\boldsymbol{q}}_0 = \boldsymbol{0} & (9) \end{aligned}$$

Baerlocher et. al [4]

Since Siciliano introduced it in recursive form, however, the null projection matrix $\boldsymbol{N}$ isn't. Therefore, Baerlocher introduced a recursive formula to enhance the efficiency of the algorithm through a recursive computation of the projector:

$$\begin{aligned} \hat{\boldsymbol{N}}_i &= \hat{\boldsymbol{N}}_{i-1} - (\boldsymbol{J}_i\hat{\boldsymbol{N}}_{i-1})^{\#}(\boldsymbol{J}_i\hat{\boldsymbol{N}}_{i-1}) \\ \hat{\boldsymbol{N}}_0 &= \boldsymbol{I} \end{aligned} \tag{10}$$

S. Chiaverini et. al [5]

Chiaverini also proposed task-priority formulation as follows:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\#}\dot{\boldsymbol{x}}_1 + (\boldsymbol{I} - \boldsymbol{J}_1^{\#}\boldsymbol{J}_1)[\boldsymbol{J}_2^{\#}\dot{\boldsymbol{x}}_2] \tag{11}$$

Compared to (4), algorithmic singularities are absent, but there is a greater tracking error for the secondary task. Therefore, he introduces damping:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\#\lambda}\dot{\boldsymbol{x}}_1 + (\boldsymbol{I} - \boldsymbol{J}_1^{\#\lambda}\boldsymbol{J}_1)[\boldsymbol{J}_2^{\#\lambda}\dot{\boldsymbol{x}}_2] \tag{12}$$

this potentially causes some interference between the tasks, when the primary task is approaching a kinematic singularity.

Kim, Lee & Ahn et. al [6]

The basic idea is to compute incremental joint positions based on operational space position errors and add them to the current joint positions. This is done using null-space task prioritization as follows:

$$\begin{aligned} \Delta\boldsymbol{q}_1 &= \boldsymbol{J}_1^{\dagger}(\boldsymbol{x}_1^{\text{des}} - \boldsymbol{x}_1) \\ \Delta\boldsymbol{q}_2 &= \Delta\boldsymbol{q}_1 + \boldsymbol{J}_{2|\text{pre}}^{\dagger}(\boldsymbol{x}_2^{\text{des}} - \boldsymbol{x}_2 - \boldsymbol{J}_2\Delta\boldsymbol{q}_1) \\ &\ \vdots \\ \Delta\boldsymbol{q}_i &= \Delta\boldsymbol{q}_{i-1} + \boldsymbol{J}_{i|\text{pre}}^{\dagger}(\boldsymbol{x}_i^{\text{des}} - \boldsymbol{x}_i - \boldsymbol{J}_i\Delta\boldsymbol{q}_{i-1}) \end{aligned} \tag{13}$$

where $\{\cdot\}^{\dagger}$ denotes an SVD-based pseudo-inverse operator in which small singular values are set to 0.

$$\begin{aligned} \boldsymbol{q}^d &= \boldsymbol{q} + \Delta\boldsymbol{q} & (14)\\ \dot{\boldsymbol{q}}^d &= \dot{\boldsymbol{q}}_{i-1}^d + \boldsymbol{J}_{i|\text{pre}}^{\dagger}(\dot{\boldsymbol{x}}_i^{\text{des}} - \boldsymbol{J}_i\dot{\boldsymbol{q}}_{i-1}^d) & (15)\\ \ddot{\boldsymbol{q}}^d &= \ddot{\boldsymbol{q}}_{i-1}^d + \boldsymbol{J}_{i|\text{pre}}^{\dagger}(\ddot{\boldsymbol{x}}_i^{\text{des}} - \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} - \boldsymbol{J}_i\ddot{\boldsymbol{q}}_{i-1}^d) & (16) \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{J}_{i|pre} &= \boldsymbol{J}_i\boldsymbol{N}_{i-1} \\ \boldsymbol{N}_{i-1} &= \boldsymbol{N}_0\boldsymbol{N}_{1|0}\cdots\boldsymbol{N}_{i-1|i-2} \\ \boldsymbol{N}_{i|i-1} &= \boldsymbol{I} - \boldsymbol{J}_{i|i-1}^{\dagger}\boldsymbol{J}_{i|i-1} \\ \boldsymbol{N}_0 &= \boldsymbol{I} - \boldsymbol{J}_c^{\dagger}\boldsymbol{J}_c \end{aligned} \tag{17}$$

He asserts that this approach is more concise and results in a lower computation load for similar control specification, compared to [3, 7].

TL;DR

Several approaches exist wishing to choose the right method to control redundant manipulators or multi-body legged robots.

References

[1] Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15, 1987.

[2] Michael Mistry, Jun Nakanishi, and Stefan Schaal. Task space control with prioritization for balance and locomotion. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 331–338. IEEE, 2007.

[3] Siciliano B Slotine and B Siciliano. A general framework for managing multiple tasks in highly redundant robotic systems. In *proceeding of 5th International Conference on Advanced Robotics*, volume 2, pages 1211–1216, 1991.

[4] Paolo Baerlocher and Ronan Boulic. Task-priority formulations for the kinematic control of highly redundant articulated structures. In *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*, volume 1, pages 323–329. IEEE, 1998.

[5] Stefano Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.

[6] Donghyun Kim, Jaemin Lee, Junhyeok Ahn, Orion Campbell, Hochul Hwang, and Luis Sentis. Computationally-robust and efficient prioritized whole-body controller with contact constraints. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[7] Ludovic Righetti and Stefan Schaal. Quadratic programming for inverse dynamics with optimal distribution of contact forces. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 538–543. IEEE, 2012.